

Information Retrieval

WS 2017 / **2018**

Lecture 14, Tuesday February 6th, 2018
(Course Evaluation, Exam, Work at our Chair)

Prof. Dr. Hannah Bast
Chair of Algorithms and Data Structures
Department of Computer Science
University of Freiburg

Overview of this lecture

■ Organizational

- Your experiences with ES13 NLP, POS-Tagging
- Official evaluation results + discussion
- Infos about the exam when, where, how, tasks
- Work at our chair how, projects, next courses

The lecture today will be streamed **live on YouTube**

In YouTube, just search: [information retrieval lecture 14](#)

■ Summary / excerpts

- The mathematics is somewhat complex, but the algorithm is rather simple to implement
- "I wouldn't have thought that NLP is so much fun."
- "Really cool, my life has changed when I understood HMM."
- "It's interesting to see how accurate it is."
- Two of you even implemented a web app (and Claudius, too)
 - "Obviously I had to do the HTTP server variant, I hope it's pretty enough"
- Does the experiences.txt count and what is the purpose?
- "Oh man, what will I be doing without IR exercises now..."

<https://drive.google.com/file/d/1JPALrwiHYulBdHr0Ghlqx5fCM4aiGlfu/view>

■ Participants

- Registered for exam: **76** ... last year: **77**
- Participated in the evaluation: **78** ... **great !**
 - 24** x Bachelor, **47** x Master, **7** x Other
 - 65** x Informatik, **11** x ESE, **2** x Other
- Nominations for teaching award: **67** ... **thanks a lot !**
- In the following, a summary of your feedback
- You find **all** the details [linked on the course Wiki](#)

■ Style of the course

- **Learned** a lot: 67% fully agree, 31% agree, 2% ok
- **Explained** well: 73% fully agree, 27% agree
- **Activity** asked for: 76% fully agree, 21% agree, 4% ok
- **Level** of contents: 48% appropriate, 51% high, 1% too high
- **Quality** overall: 73% very good, 27% good
- You liked: very good and clear explanations, competent, very motivated and motivating, very interesting / not boring, combination of theory and practice, great exercise sheets, live coding, recordings, quick answers on forum almost 24/7, exceptional effort, quite entertaining
- Criticism / suggestions: see slide 9

Results Course Evaluation 3/9

■ Student's effort

- Effort relative to ECTS ... 1 = very high, 5 = very low

22% x 1 41% x 2 36% x 3 0% x 4 1% x 5 this course

30% x 1 45% x 2 25% x 3 0% x 4 0% x 5 last year

12% x 1 31% x 2 51% x 3 5% x 4 1% x 5 department
average

- So still more work than other CS lectures

It was also an occasional comment in the written feedback

Last year, it was a more frequent comment, so apparently the exercise sheets were less work and/or more enjoyable this time around

At least we worked pretty hard to make them (even) better

■ Materials and Online Support / Tutors

- **Materials** helpful: 72% fully agree, 22% agree, 5% ok
- Consumed lecture by **presence or video recordings**:
 - 14% pres, 39% rec, 38% both, 10% other this course
 - 23% pres, 52% rec, 21% both, 4% other last year
 - 34% pres, 15% rec, 19% both, 32% other dep average
- Video recordings:

Great quality + this time even with live streaming

Thank you very much **Frank** (technical setup), **Claudius** (live stream), and **Alexander Monneret** (editing)

■ Assistant and Tutors

- **Assistant** (Claudius) ... synonyms: Claus, Mr. Claudius

Fantastic support behind the scenes, in particular for:
the various datasets, the master solutions, the exercise
sheets in general, the forum ... **thank you very much**

"Prof. Hannah and Mr. Claudius were always well-prepared
for us. They are my super-heroes! (my tutor, as well!)"

- **Tutors** (Patrick, Natalie, Daniel, Johannes, Danny, Simon):

Very friendly and helpful, fast (mostly) and detailed feedback,
competent, fair, always available

Many of you explicitly thanked the whole team of the course

■ Criticism / Suggestions

- Some of the exercise sheets were too much work
- Changes to exercise sheets after they were issued
- Problems with the pen, hard to read
- Formulas written in PowerPoint are ugly
- Handwritten notes and code is too small ... back row?
- Adapting unit tests to language can be cumbersome
- Livestream with live chat / comment function
- Web App sheets: many people liked it, a few didn't
- Some of the tutors were sometimes very (too?) strict
checkstyle errors due to last-minute changes → 0 points

■ Planned improvements **from last year**

- More theoretical sheets, in alternation with the (often more time-intensive) implementation sheet ✓
- Consider the many notes taking after each lecture ✓
- Clarify that test cases are mandatory, but also clarify that they do not have to be implemented 1-1 ✓
- Clarify in advance which programming languages are appropriate for which exercise sheet ✓
- Finish Python cheat sheet and make it available early ✓
- Offer live tutorials again, until nobody comes anymore (last year: one live tutorial, with meagre attendance) ✗

- Planned improvements **for next year** part 1/2
 - Keep the mix theoretical sheets / implementation sheets
 - Try to reduce the workload of the sheets further
 - New touchscreen with new pen that works perfectly, and nicer formulas in PowerPoint, if possible
 - Livestream via YouTube ... maybe
 - Offer one or two live tutorials (after theory lecture)
 - Finish Python cheat sheet and make it available early
 - Clarify that master solutions are for personal use only
 - Clear rule on what to write in the copyright notice
 - Consider the notes I took on how each lecture went

- Planned improvements **for next year** part 2/2
 - Automatic subscription to Daphne Announcement subforum for all students registered to the course
 - Upon Daphne registration, ask for preferred prog. language, and don't sort people by their first name (three times Niklas)
 - More incentive/award for writing modular (non-spaghetti) code + say which code is re-used in later sheets
 - Better overview over points (and percentage) in Daphne
 - More help with C++ timeout and std::wstring
 - Provide instruction on how to enlarge the RAM of a VM on the Wiki (along with the link for the download)

- Where, when, what to bring

- Oral exams (B.Sc. Computer Science students only):

Wednesday+Friday, February 21+23, 30min in 51-02-28

- Written exam (all other participants):

Monday, February 19, 14:00 – 16:30 in 101-036 + 082-006

- Please bring: student id, colored pens, brain

student id: make sure you look like your photo or vice versa

colors: greatly improves readability for examples / drawings

brain: greatly improves quality of answers in general

■ Type of exam

- The written exam is **open book**

That is, you can bring books, paper, etc ... but please be ecological when printing out slides + good for your karma

Electronic devices of any kind are obviously not allowed

- In the oral exam, ask us if you are missing some detail

But there is no time to start understanding things for the first time, during neither the oral nor the written exam

- There will be a sub-forum for questions about the exam

Answer speed will be slightly slower in the semester break

Don't ask all your questions in the night before the exam

■ Types of questions

- **Type 1:** Do algorithm (or variant thereof) by example, like we often did in the lecture **see colored pens**
- **Type 2:** Implement a small function (in Python, Java or C++) **small indeed, at most 10 lines per task**
- **Type 3:** Small calculations or proofs **see brain**
- In general: the emphasis is on **understanding**, not on learning things by heart
- Important: all the contents + insights from the **exercises** are (very) relevant for the exam

■ Preparation

- To check whether you understood something:

Put away the material from the lecture and try to write it down / prove it **in your own words / formalism**

There is no point in learning the individual steps of a proof or argument by heart ... it doesn't work that way

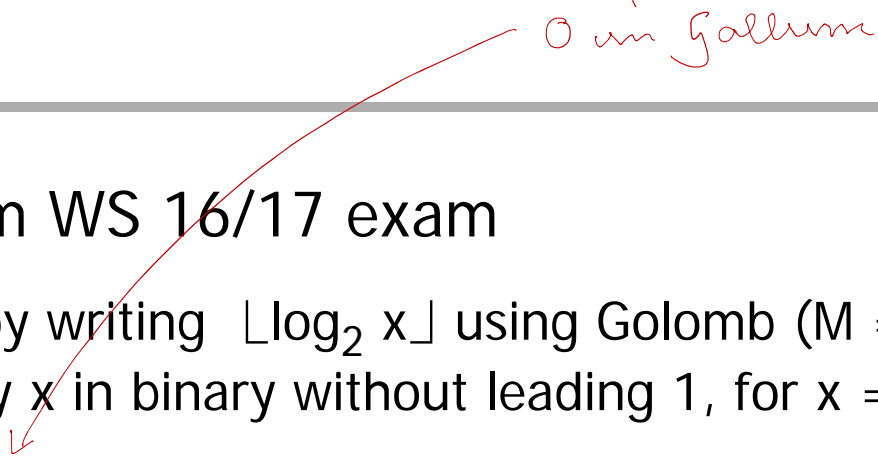
- Once the basic stuff from the lecture is understood, the best preparation is to do all the exercise sheets

If you work your way through all the exercise sheets (yourself), there is no way you can fail the exam

- Also: work through the old exams (linked on the Wiki)

■ Task 2.1 from WS 16/17 exam

- Encode x by writing $\lfloor \log_2 x \rfloor$ using Golomb ($M = 4$) followed by x in binary without leading 1, for $x = 1, \dots, 10$



1	100.
2	101.0
3	101.1
4	110.00
5	110.01
6	110.10
7	110.11
8	111.000
9	111.001
10	111.010

■ Task 3.4 from WS 16/17 exam

- Write a function that counts the number of chars in a valid UTF-8 sequence (given as a byte array)

```
def num_chars(bytes):  
    count = 0  
    for byte in bytes:  
        if byte & 128 + 64 != 128:  
            count += 1  
    return count
```

Handwritten red annotations:
A red circle highlights the condition `if byte & 128 + 64 != 128:`.
A red line points from the text `11000000b` to the `128` in the condition.
A red line points from the text `10000000b` to the `64` in the condition.

■ Task 6.3 from WS 16/17 exam

$$[x \cdot \ln x]' = 1 \cdot \ln x + x \cdot \frac{1}{x}$$

- What is the maximum entropy of a probability distribution with probabilities p_1, \dots, p_n (use Lagrangian optimization)

$$H = - \sum_{i=1}^n p_i \cdot \log_2 p_i \quad \leftarrow \text{maximize this}$$

$$\text{minimize } \sum_{i=1}^n p_i \cdot \ln p_i$$

$$L := \sum_{i=1}^n p_i \cdot \ln p_i - \lambda \left(\sum_{i=1}^n p_i - 1 \right) \quad \text{MIN} \quad \leftarrow$$

$$\frac{\partial L}{\partial p_i} = \ln p_i + 1 - \lambda \stackrel{!}{=} 0 \quad ; \quad \frac{\partial^2 L}{\partial p_i^2} = \frac{1}{p_i} > 0$$

$$\Rightarrow p_i \text{ all equal} \Rightarrow p_i = \frac{1}{n}$$

$$\Rightarrow H = - \sum_{i=1}^n \frac{1}{n} \cdot \log_2 \frac{1}{n} = \log_2 n$$

and we have many
really interesting
and large **datasets**

■ How we work

- We solve practically relevant (usually hard) problems

Route Planning, Transit Maps, Search As You Type,
Semantic Search, Question Answering, PDF extraction, ...

- We use theory as a (vital / essential) tool

All our work has a solid theoretical understanding, otherwise
solving complex problems remains hacking and guesswork

- We make our software + results available to the public

This requires an effort to write good software, good
documentation, nice user interfaces, and so on ...

All three aspects were clearly visible in this course, too !

■ Supervision

- Similarly as in the lecture:

Very good infrastructure + support, but apart from that you can (and are supposed to) work very independently

Team work is, of course, highly desirable and encouraged

Great for enthusiastic people who care about practical stuff and who want to get things done

■ Machine Learning

- We are building more and more on **machine learning** to solve our problems

Not because it's fashionable ... but because it's practical

It's quite obvious that learning is the future for problems like natural language understanding

- You have seen a few learning algorithms in this lecture:

k-means, Naive Bayes, LSI, Hidden Markov Models, ...

For a few years now, we also use **deep learning**

The complexity lies not so much in the algorithms, but in understanding how and why they work how well

■ Current projects and demos

- Route planning (part of Google Maps) [demo](#)
- World-wide public transit visualization (Travic) [demo](#)
- Automatic drawing of nice transit maps [demo](#)
- Large-scale SPARQL+Text search (QLever) [demo](#)
- Search-as-you-type semantic search (Broccoli) [demo](#)
- Question Answering (Aqqu) [demo](#)
- Question Completion [demo](#)
- Text extraction from PDF (Icecite) [paper](#)

■ Upcoming courses

- **Programming in C++** ... in the SS 2018

2nd semester B.Sc. Info + 4th semester B.Sc. ESE

- **Information Retrieval** ... in the WS 2018/2019

You know it ... become a tutor if you do a great exam !

- **Algorithms and Data Structures** ... in the SS 2019

Basic course for 2nd semester B.Sc. Informatik students

- **B.Sc. / M.Sc. projects or theses**

Offered all the time, read the instructions on our **Wiki**
and then just ask (if no response, ask again please)

It helps if you attended one my lectures, with good grade